Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Claims 1-21 (previously canceled).

22. (previously presented) A method, comprising:

compressing an immediate operand associated with a current instruction;

storing the compressed operand in a selected one of a plurality of fixed-length operand fields, wherein each of the operand fields is associated with one of a plurality of program instructions;

wherein the plurality of program instructions includes the current instruction and an adjacent instruction.

23. (previously presented) The method of claim 22, wherein:

the adjacent instruction is a previous instruction; and

storing the compressed operand further comprises storing the compressed operand in the operand field associated with the previous instruction.

24. (previously presented) The method of claim 23, further comprising:

storing in a control field associated with the previous instruction a value to indicate that the compressed operand should be ignored when executing the previous instruction.

25. (previously presented) The method of claim 22, wherein:

storing the compressed operand further comprises storing the

compressed operand in the operand field associated with the current instruction.

26. (previously presented) The method of claim 25, further comprising:

storing in a control field associated with the current instruction a value to indicate sign extension compression.

27. (previously presented) The method of claim 22, wherein:

the adjacent instruction is a next instruction; and

storing the compressed operand further comprises storing the compressed operand in the operand field associated with the next instruction.

28. (previously presented) The method of claim 27, further comprising:

storing in a control field associated with the next instruction a value to indicate that the compressed operand should be ignored when executing the next instruction.

29. (previously presented) The method of claim 22, wherein compressing further comprises:

compressing using sign extension.

30. (previously presented) The method of claim 22, wherein:

the plurality of program instructions includes the current instruction, a previous instruction and a next instruction.

31. (previously presented) The method of claim 30, wherein storing the compressed operand further comprises:

storing the compressed operand in the operand field for the previous instruction if the data field of the previous instruction is available.

32. (previously presented) The method of claim 30, wherein storing the compressed operand further comprises:

storing the compressed operand in the operand field for the current instruction if the operand field of the previous instruction is unavailable and the operand field of the current instruction is available.

33. (previously presented) The method of claim 30, wherein storing the compressed operand further comprises:

storing the compressed operand in the operand field of the next instruction if the operand field of the previous instruction is unavailable and the operand field of the current instruction is unavailable and the operand field of the next instruction is available.

34. (previously presented) The method of claim 30, wherein storing the compressed operand further comprises:

generating a nop instruction if the operand field of the previous instruction is unavailable and the operand field of the current instruction is unavailable and the operand field of the next instruction is unavailable; and

storing the compressed operand in an operand field associated with the nop instruction.

35. (previously presented) The method of claim 34, wherein generating the nop instruction further comprises:

inserting the nop instruction between the current instruction and the next instruction.

36. (previously presented) A method, comprising:

storing one portion of an immediate operand for a current instruction in a fixed-length operand field associated with the current instruction; and

storing a remaining portion of the immediate operand for the current instruction in a fixed-length operand field of an instruction adjacent to the current instruction;

wherein the length of the immediate operand is Y bits and the length of the

operand fields for the current instruction and the adjacent instruction is less than Y bits.

37. (previously presented) The method of claim 36, wherein the length of the operand fields for the current instruction and the adjacent instruction is Y/2.

38. (previously presented) The method of claim 36, wherein the adjacent instruction is a previous instruction.

39. (previously presented) The method of claim 38, further comprising:

storing in a control field associated with the previous instruction a value to indicate backward scavenging compression.

40. (previously presented) The method of claim 36, wherein the adjacent instruction is a next instruction.

41. (previously presented) The method of claim 40, further comprising:

storing in a control field associated with the next instruction a value to indicate forward scavenging compression.

42. (withdrawn) A method, comprising:

determining if at least a portion of the bit pattern of an immediate operand associated with a current instruction is identical to a bit pattern stored in an operand field associated with an adjacent instruction; and

if so, indicating in a control code field that the current instruction and the adjacent instruction should share the stored bit pattern.

43. (withdrawn) The method according to claim 42, wherein:

the adjacent instruction is a previous instruction.

44. (withdrawn) The method according to claim 43, further comprising:

clearing an operand field associated with the current instruction.

45. (withdrawn) The method according to claim 43, further comprising:

storing the bit pattern in an operand field associated with the current instruction; and

clearing the operand field of the previous instruction.

46. (withdrawn) The method according to claim 42, wherein indicating in a control code field that the current instruction and the adjacent instruction should share the stored operand further comprises:

indicating, in a control code field associated with the current instruction, that the immediate operand associated with the adjacent instruction is to be used to execute the current instruction.

47. (withdrawn) The method according to claim 42, wherein the adjacent instruction is a next instruction.

48. (withdrawn) The method according to claim 47, further comprising:

clearing an operand field associated with the current instruction.

49. (withdrawn) The method according to claim 47, further comprising:

storing the bit pattern in an operand field associated with the current instruction; and

clearing the operand field associated with the next instruction.

50. (withdrawn) The method according to claim 49, wherein indicating in a control code field that the current instruction and the adjacent instruction should share the stored operand further comprises:

indicating, in a control code field associated with the adjacent instruction, that the immediate operand associated with the current instruction is to be used to execute the adjacent instruction.